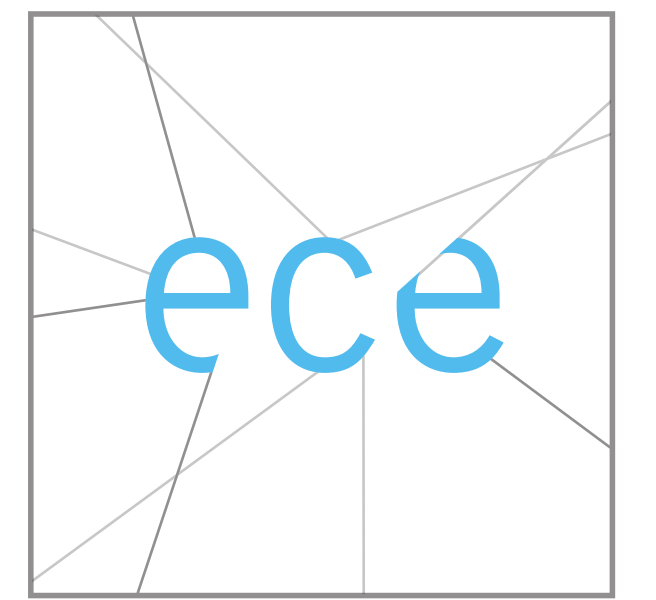




SmartJS: Dynamic and Self-Adaptable Runtime Middleware for Next-Generation IoT Systems



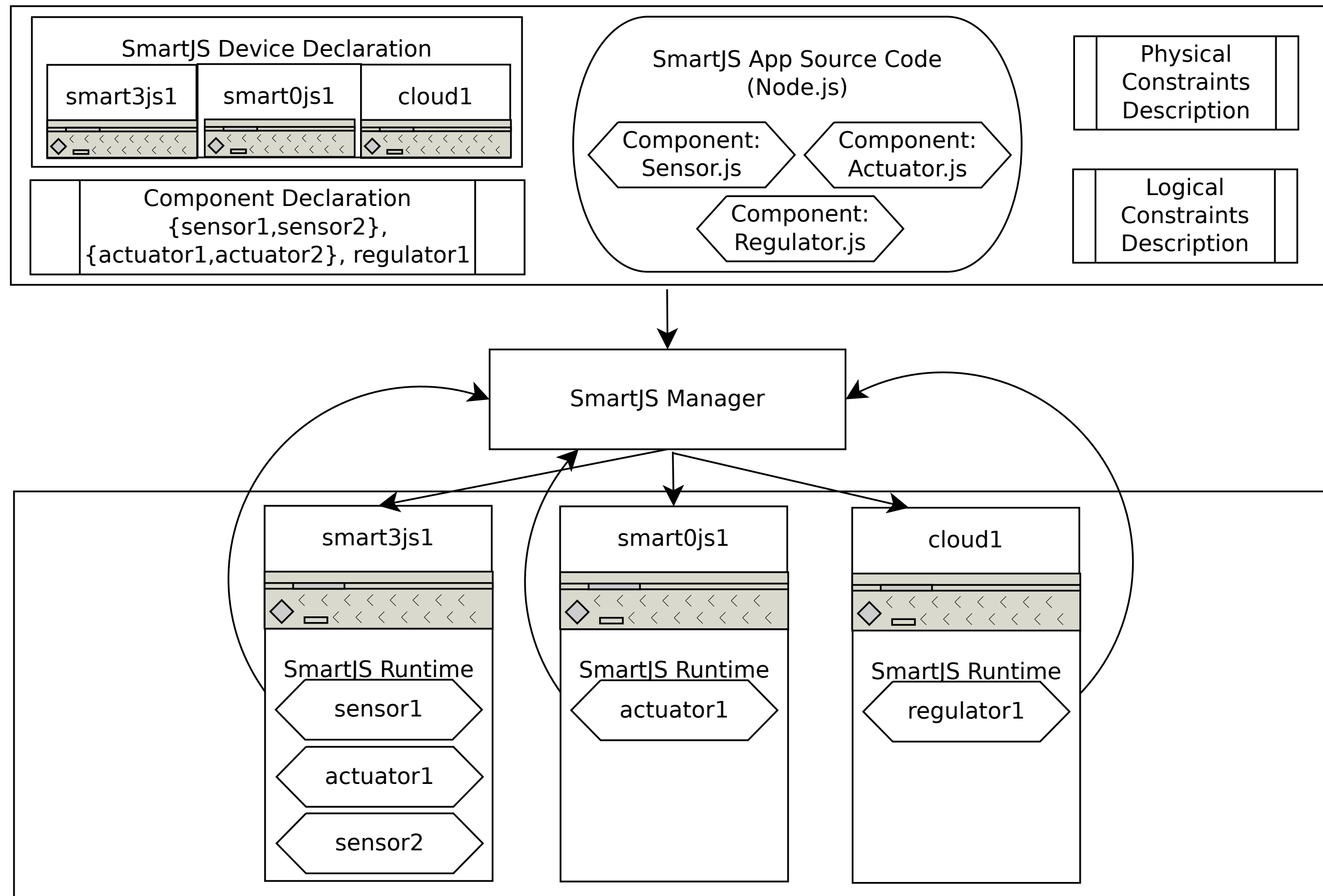
SPLASH 2017, Vancouver, BC Canada

Julien Gascon-Samson, Mohammad Rafiuzzaman and Karthik Pattabiraman
Department of Electrical and Computer Engineering, University of British Columbia, Canada

Motivation

SmartJS: a rich Javascript-based self-adaptable runtime environment which features a **universal programming API**, a **comprehensive monitoring framework** and an **ubiquitous communication substrate** for engineering and developing **dependable, scalable, adaptable large-scale IoT systems**.

System Architecture



Physical and Logical Constraints

Physical Constraints - Device-Specific

- CPU (Workload units)
- RAM
- Bandwidth: incoming/outgoing, towards other nodes
- Latency (in relation to other nodes)

Logical Constraints - Component-Specific

- CPU (Workload units)
- RAM
- Bandwidth: incoming/outgoing, towards other components
- Latency (in relation to other components)

Scheduling - Predictive Model

	Feature 1	Feature 2	Feature d	Execution Time
node 1	xxx	xxx	xxx	xxx
node 2	xxx	xxx	xxx	xxx
node 3	xxx	xxx	xxx	xxx
...
node 10	xxx	xxx	xxx	xxx
node 11	xxx	xxx	xxx	xxx
node 12	xxx	xxx	xxx	xxx
...
node 15	xxx	xxx	xxx	xxx

Testing Input features and Output based on those Input Features

Training Samples where both I/O is present and Output is generated based on the input features

Predicted Output values based on the Input Features and then compared with the actual output values to see the accuracy of the prediction model

Writing a SmartJS Application

```

1 // ...
2
3 // Connect
4 pubsub.connect(function() {
5
6     // Repeat every second
7     setInterval(function() {
8
9         // Read temperature from GPIO pin
10        var temperature = GPIO.readPin(12);
11
12        // Publish temperature
13        pubsub.publish("smartsensor/temperature", {
14            id: mySensorId,
15            temperature: temperature
16        });
17
18    }, 1000);
19
20 });

```

sensor.js

```

1 // ...
2
3 // Connect
4 pubsub.connect(function() {
5
6     // Subscribe to temperature messages
7     pubsub.subscribe("smartsensor/temperature", function(d) {
8
9         if (d.temperature > threshold) {
10            pubsub.publish("smartsensor/actuation", {
11                id: d.id,
12                powerVariation: -5
13            });
14        } else if (d.temperature < threshold) {
15            pubsub.publish("smartsensor/actuation", {
16                id: d.id,
17                powerVariation: 5
18            });
19        }
20    });
21
22 });

```

regulator.js

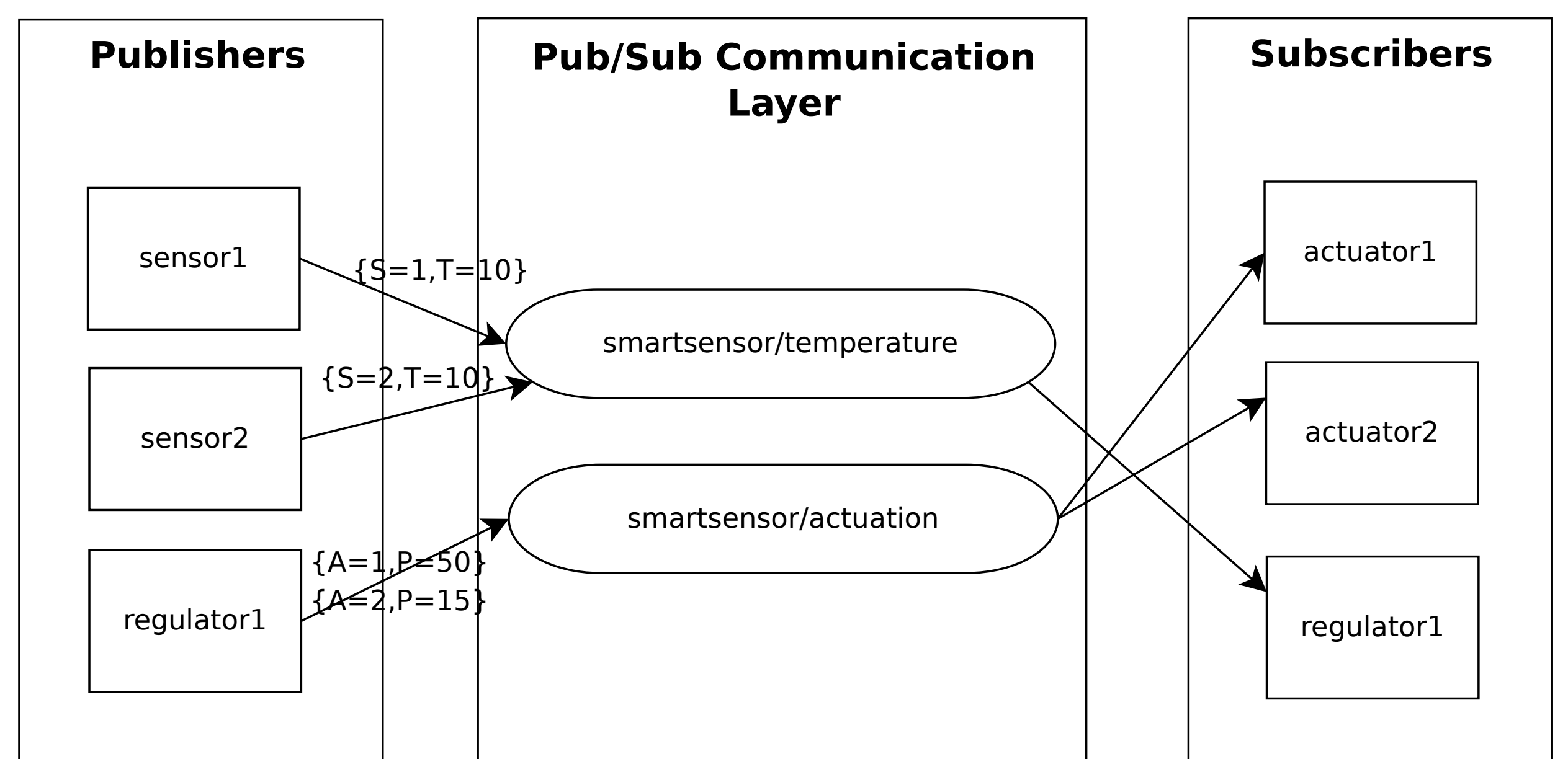
```

1 // ...
2
3 // Connect
4 pubsub.connect(function() {
5
6     // Subscribe to change power topic to receive commands
7     // from the manager
8     pubsub.subscribe("smartsensor/actuation", function(d) {
9
10        // Adjust the power by writing to GPIO pin
11        GPIO.writePin(14, d.powerVariation);
12
13    });
14
15 });

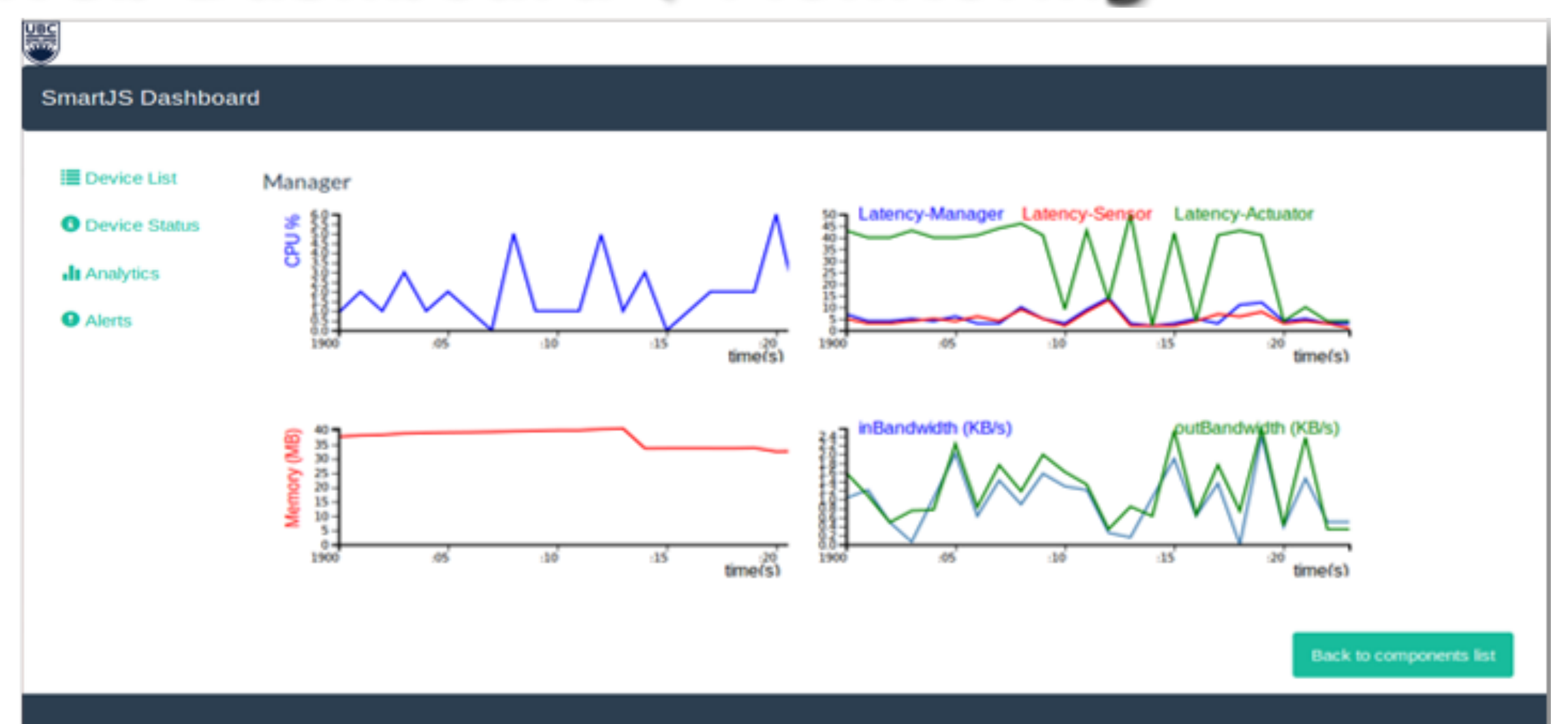
```

actuator.js

Publish-Subscribe Paradigm



Web Dashboard / Monitoring



References

- [1] 2014. Intel XDK. (2014). <https://software.intel.com/en-us/xdk>
- [2] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Conference on Security*. USENIX, Berkeley, USA.
- [3] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. 2003. The Many Faces of Publish/Subscribe. *ACM Comput. Surv.* 35, 2 (2003), 114–131.
- [4] Evgeny Gavrin, Sung-Jae Lee, Ruben Ayrapetyan, and Andrey Shitov. 2015. Ultra Lightweight JavaScript Engine for Internet of Things. In *SPLASH Companion 2015*. ACM, New York, NY, USA, 19–20.
- [5] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, and Jesus Alonso-Zarate. 2015. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud Computing* 3, 1 (2015), 11–17.
- [6] Jin-woo Kwon and Soo-Mook Moon. 2017. Web Application Migration with Closure Reconstruction. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. Republic and Canton of Geneva, Switzerland, 133–142.
- [7] James Teng Kin Lo, Eric Wohlstadt, and Ali Mesbah. 2013. Imagen: Runtime Migration of Browser Sessions for Javascript Web Applications. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*. ACM, New York, NY, USA, 815–826.